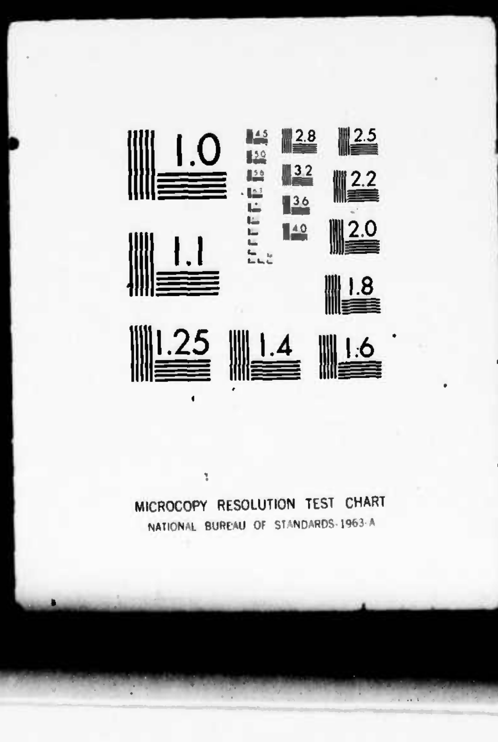
AD-A095 539 HARVARD UNIV CAMBRIDGE MASS AIKEN COMPUTATION LAB F/G 12/1
MINIMUM S-T CUT OF A PLANAR UNDIRECTED NETWORK IN O(N LOG2(N)) --ETC
1980 J H REIF N00014-80-C-0647
UNCLASSIFIED TR-20-80 NL

UNCLAS

1 OF 1

AD
A095539



HARVARD UNIVERSITY
CENTER FOR RESEARCH IN COMPUTING TECHNOLOGY

January 13, 1981

Chief of Naval Research
Code 437
800 North Quincy Street
Arlington, VA 22217

Dear Sir:

Enclosed are the papers "Minimum S-T Cut of a Planar Undirected Network in $O(n \log^2(n))$ Time", The Complexity of Provable Properties of First Order Theories", and "On Probabilistic and Symmetric Parallel Computations". This work was supported by ONR contract N00014-80-C-0647.

Sincerely,

John H. Reif
Assistant Professor
of Computer Science

JHR:bm
cc: ONR Branch Office; Eastern/Central Region
M. Kelley, ONR Representative
Naval Research Laboratory
DDC, Alexandria, VA

AIKEN COMPUTATION LABORATORY
Cambridge, Mass. 02138

2

6

MINIMUM S-T CUT OF A PLANAR UNDIRECTED NETWORK

IN $O(n \log^2(n))$ TIME

by

12 22

10 John H. Reif

14 TR-20-80

11 1980

15 N00014-80-C-0647, NSF-MCS79-21024

MINIMUM S-T CUT OF A PLANAR UNDIRECTED NETWORK IN $O(n \log^2(n))$ TIME

by

John H. Reif

Aiken Computation Lab., Harvard University, Cambridge, MA 02138.
This work was supported in part by the National Science Foundation grant NSF-MCS79-21024 and the Office of Naval Research grant N00014-80-C-0647.

Minimum s-t Cut of a Planar Undirected Network in $O(n \log^2(n))$ Time

Summary. Let N be a planar undirected network with distinguished vertices s, t, a total of n vertices, and each edge labeled with a positive real (the edge's cost) from a set L. This paper presents an algorithm for computing a minimum (cost) s-t cut of N.

For general L, this algorithm runs in time $O(n \log^2(n))$ time on a (uniform cost criteria) RAM. For the case L contains only integers $\leq n^{O(1)}$, the algorithm runs in time $O(n \log(n) \log\log(n))$. Our algorithm also constructs a minimum s-t cut of a planar graph (i.e., for the case $L = \{1\}$ in time $O(n \log(n))$.

The fastest previous algorithm for computing a minimum s-t cut of a planar undirected network [Gomory and Hu, 1961] and [Itai and Shiloach, 1979] has time $O(n^2 \log(n))$ and the best previous time bound for minimum s-t cut of a planar graph [Cheston, Probert, and Saxton, 1977] was $O(n^2)$.

1. Introduction

The importance of computing a minimum s-t cut of a network is illustrated by Ford and Fulkerson's [1962] Theorem which states that the value of the minimum s-t flow of a network is precisely the minimum s-t cut.

The best known algorithms [Galil, Naamad 1979],[Shiloach, 1978] for computing the max flow or minimum s-t cut of a *sparse directed or undirected network* (with n vertices and $O(n)$ edges) has time $O(n^2 \log^2(n))$.

This paper is concerned with a *planar undirected network* N, which occurs in many practical applications.

Ford and Fulkerson [1956] have an elegant minimum s-t cut algorithm for the case N is (s,t)-*planar* (both s and t are on the same face) which efficiently implemented by [Gomory and Hu, 1961] and [Itai and Shiloach, 1979] has time $O(n \log(n))$.

Moreover, $O(n)$ executions of their algorithm suffices to compute the minimum s-t cut of an *arbitrary planar network* in total time $O(n^2 \log(n))$. Also, [Cheston, Probert, Saxton, 1977] have an $O(n^2)$ algorithm for the minimum s-t cut of a planar graph.

A key element of the [Ford and Fulkerson, 1956] algorithm for (s,t)-planar networks was an efficient reduction to finding a minimum cost path between two vertices in a sparse network. [Dijkstra, 1959] gives an algorithm for a generalization of this problem (to find a minimum cost path from a fixed "source" vertex s to each other vertex). Dijkstra's algorithm may be implemented (see [Aho, Hopcroft and Ullman, 1974]) in time $O(Q_L(n))$ for

a sparse network with n vertices, L is the set of non-negative reals labeling the edges, and $Q_L(n)$ is an upper bound on the time to maintain a queue of $O(n)$ elements with costs from L, and with $O(n)$ insertions and deletions. For the general case, $Q_L(n) = O(n \log(n))$ (see [Hopcroft and Ullman, 1974]). For the special case L is a set of positive integers $\leq n^{O(1)}$ [Boas, Kaas and Zijlstra, 1977], $Q_L(n) = O(n \log\log(n))$. It is obvious that if $L = \{1\}$, $Q_L(n) = O(n)$.

Our algorithm for computing the minimum s-t cut of a planar undirected network has time $O(Q_L(n)\log(n))$. This algorithm also utilizes an efficient reduction to minimum cost path problems. Our fundamental innovation is a divide and conquer approach for cuts on the plane.

The paper is organized as follows:

The next section gives preliminary definitions of graphs, networks, min cuts, and duals of planar networks. Section 3 gives the Ford-Fulkerson Algorithm for (s,t)-planar graphs.

Section 4 gives an efficient algorithm for minimum cut graphs containing a given face. Our divide and conquer approach is described and proved in Section 5. Section 6 presents our algorithm for minimum s-t cuts of planar networks.

Finally, Section 7 concludes the paper.

2. Preliminary Definitions

2.1 Graphs

Let a *graph* $G = (V,E)$ consist of a *vertex set* V and a collection of *edges* E. Each edge $e \in E$ *connects* two vertices $u, v \in V$ (edge e is a *loop* if it connects identical vertices). We let $\underline{e = \{u,v\}}$ denote edge e connects u and v. Edges e, e' are *multiple* if they have the same connections.

Let a *path* be a sequence of edges $p = e_1,\ldots,e_k$ such that $e_i = \{v_{i-1},v_i\}$ for $i = 1,\ldots,k$ (we say p *traverses* vertices $v_0,\ldots,v_k$). Let p be a *cycle* if $v_0 = v_k$ (cycles containing the same edges are considered identical). A path p' is a *subpath* of p if p' is a subsequence of p.

Let G be a *standard graph* if G has no multiple edges nor loops. Generally we let $\underline{n}$ be the number of vertices of graph G. G is *sparse* if the number of edges is $O(n)$. If G is planar, then by Euler's Theorem G is sparse and contains at most $6n-12$ edges.

2.2 Networks

Let an *undirected network* $N = (G,c)$ consists of a graph $G = (V,E)$ and a mapping c from E to the positive reals. For each edge $e \in V$, $c(e)$ is the *cost of* e. For any edge set $E' \subseteq E$, let $c(E') = \Sigma_{e \in E'} c(e)$. Let the *cost* of path $p = e_1,\ldots,e_k$ be $c(p) = \Sigma_{i=1}^{k} c(e_i)$. Let a path p from vertex u to vertex v be *minimum* if $c(p) \le c(p')$ for all paths p' from u to v.

Let $N = (G,c,s,t)$ be a *standard network* if (G,c) is an undirected network, with $G = (V,E)$ a standard graph, and s, t are distinguished vertices of V (the *source*, *sink* respectively).

2.3 Min Cuts and Flows in Networks

Let $N = (G,c,s,t)$ be a standard network with $G = (V,E)$.

An edge set $X \subseteq E$ is a s-t *cut* if $(V,E-X)$ has no paths from s to t. Let s-t cut X be *minimum* if $c(X) \leq c(X')$ for each s-t cut X.

A function f mapping E to the nonnegative reals is a *flow* if

(i) $\forall e \in E,\ f(e) \leq c(e)$.

(ii) $\forall v \in V-\{s,t\},\ ID(f,v) = OUT(f,v)$

where

$$IN(f,v) = \sum_{\substack{e \in E \\ v \in e}} f(e)$$

$$OUT(f,v) = \sum_{\substack{e \in E \\ v \in e}} f(e) \quad .$$

The *value* of the flow f is

$$OUT(f,s) - IN(f,t) \quad .$$

The following motivates our work on minimum s-t cuts:

Theorem 1. [Ford and Fulkerson, 1962]. The maximum value of any flow is the cost of a minimum s-t cut.

2.4 Planar Networks and Duals

Let $G = (V,E)$ be a planar standard graph, with a fixed embedding on the plane. Each connected region of G is a *face* and has a corresponding cycle of edges which it borders. For each edge $e \in E$, let $D(e)$ be the

corresponding *dual edge* connecting the two faces bordering e.

Let $D(G) = (\mathcal{F}, D(E))$ be the *dual graph* of G, with vertex set $\mathcal{F}$ = the faces of G, and with edge set $D(E) = \cup_{e \in E} D(e)$.

Note that the dual graph is not necessarily standard (i.e., it may contain multiple edges and loops), but is planar.

Let a cycle q of $D(G)$ be a *cut-cycle* if the region bounded by q contains exactly one of s or t.

Proposition 1. D induces an isomorphism between the s-t cuts of G and the cut-cycles of $D(G)$.

Let $N = (G,c,s,t)$ be a *planar* standard network, with $G = (V,E)$ planar.

Let the *dual network* $D(N) = (D(G),D(c))$ have edge costs $D(c)$, where $D(c)(D(e)) = c(e)$ for all edges $e \in E$. (Generally we will use just c in place of $D(c)$ where no confusion with result.) For each face $F \in \mathcal{F}$, let a cut-cyle q in $D(N)$ be F_i-*minimum* if q contains F_i and $c(q) \leq c(q')$ for all cut-cyles q' containing F_i.

Proposition 2. A minimum s-t cut has the same cost as a minimum cost cut-cycle of $D(G)$.

3. Ford and Fulkerson's Min s-t Cut Algorithm for (s,t)-Planar Networks

Let $N = (G,c,s,t)$ be a planar standard network. G (and also N) is (s,t)-*planar* if there exists a face F_0 containing both s and t. Let planar network N' be derived from N by adding on edge e_0 connecting s and t with cost ∞. Let e_0 be embedded onto a line segment from s to t in F_0, which separates F_0 into two new faces F_1 and F_2.

[Ford and Fulkerson, 1956] have an elegant characterization of the minimum s-t cut of (s,t)-planar network N.

Theorem 2. There is an isomorphism between the s-t cuts of N and the paths of $D(N')$ from F_2 to F_1 and avoiding $D(e_0)$. Furthermore, this isomorphism preserves edge costs. Therefore, the minimum s-t cuts of N correspond to the minimum cost paths in $D(N')$ from F_2 to F_1 (which avoid $D(e_0)$).

Corollary 2. A minimum cost cut of (s,t)-planar N with n vertices may be computed in time $O(Q_L(n))$, where $L = \text{range}(c)$.

Note that this implies the $O(n \log(n))$ time minimum s-t cut algorithm of [Gomory and Hu, 1961] and [Itai and Shiloach, 1979] for (s,t)-planar undirected networks, and the $O(n)$ time minimum s-t cut algorithm of [Cheston, Probert, and Saxton, 1977] for (s,t)-planar graphs.

4. An O(n log(n)) Algorithm for F-minimum Cut Cycles

Let $N = (G,c,s,t)$ be a planar standard network, with $G = (V,E)$ and $L = \text{range}(c)$. Our algorithm for minimum s-t cuts will require efficient construction of F-minimum cut cycles for certain given faces F.

Let $\mathcal{F}_s$ be the set of faces bordering s and let $\mathcal{F}_t$ be the faces bordering t. Let a $\mu(s,t)$ *path* be a minimum cost path in $D(N)$ from a face of $\mathcal{F}_s$ to a face of $\mathcal{F}_t$.

Proposition 3. Let μ be a $\mu(s,t)$ path traversing faces $F_1,\ldots,F_d$. Let q_i be a F_i-minimum cut-cycle of $D(N)$ for $i = 1,\ldots,d$. Then $D^{-1}(q_{i_0})$ is a minimum s-t cut of N, where $c(q_{i_0}) = \min\{c(q_i) \mid i = 1,\ldots,d\}$.

(Note: It is easy to compute a $\mu(s,t)$ path in time $O(Q_L(n))$. Let M be the planar network derived from $D(N)$ by adding new vertices v_s, v_t and an edge connecting v_s to each face in $\mathcal{F}_s$ and an edge connecting each face in $\mathcal{F}_t$ to v_t. Let the cost of each of these edges be 1. Let p be a minimum cost path in M from v_s to v_t. Then p, less its first and last edges, is a $\mu(s,t)$ path.)

Let μ be a $\mu(s,t)$ path traversing faces $F_1,\ldots,F_d$.

By viewing μ as a horizontal line segment with s on the left and t on the right, each edge connected to a face F_i may be considered to be connected to F_i from the *below* or *above* (or *both*).

Let μ' be a copy of μ traversing new vertices $x_1,\ldots,x_d$. Let D' be the network derived from $D(N)$ by reconnecting to x_i each edge entering F_i from above.

If p is a path of D', then a corresponding path $\hat{p}$ in $D(N)$ is constructed by replacing each edge and face appearing in μ' with the corresponding edge or face of μ. Clearly, $c(p) = c(\hat{p})$.

Theorem 3. If p is a minimum cost path connecting F_i and x_i in D', then $\hat{p}$ is a F_i-minimum cut cycle of $D(N)$.

Proof. Clearly, $\hat{p}$ is a cut-cycle of $D(N)$. Suppose $\hat{p}$ is *not* F_i-minimum. Let q be a F_i-minimum cut-cycle of $D(N)$, with $c(q) < c(\hat{p})$. Then there must be a subpath q_1 of q connecting faces F_j, F_k of μ but otherwise disjoint from μ and such that the edges of q_1 together with μ form a cut-cycle of $D(N)$ (else we can show q is not a cut-cycle).

Let μ_1 be the minimal subpath of μ containing faces F_i, F_j, and F_k. Observe that the edges of q_1 together with μ_1 form a F_i-minimum cut-cycle, else μ is not a $\mu(s,t)$ path. Let q_1' be derived from q_1 by reconnecting the last edge to x_k instead of F_k. Let μ_2 be the subpath of μ_1 connecting F_i and F_j and let μ_3 be the subpath of μ_1 connecting F_i and F_k. Also, let μ_3' be the subpath of μ' in D' corresponding to μ_3. *Then the edges of* μ_2, q_1', *and* μ_3' *form a path from* F_i *to* x_i *in* D' and with cost $c(q)$. But $c(q) < c(\hat{p}) = c(p)$ is a contradiction with the assumption that p is a minimum cost path from F_i to x_i. □

Corollary 3. There is an $O(Q_L(n))$ time algorithm to compute a F_i-minimum cut cycle for any face F_i of a $\mu(s,t)$ path in $D(N)$.

5. A Divide and Conquer Approach

Let μ be a $\mu(s,t)$ path of $D(N)$ traversing faces $F_1,\ldots,F_d$ as in Section 4. Note that any s-t cut of planar network N must contain an edge bounding on a face $F_1,\ldots,$ or F_d. Thus an obvious algorithm for computing a minimum s-t cut of N is to construct a F_i-minimum cut cycle q_i in $D(N)$ for each $i = 1,\ldots,d$. This may be done by d executions of the $O(Q_L(n))$ time algorithm of Corollary 3. Then by Proposition 3, $D^{-1}(q_{i_0})$ is a minimum s-t cut where $c(q_{i_0}) = \min\{c(q_1),\ldots,c(q_d)\}$. In the worst case, this requires $O(Q_L(n)\cdot n)$ total time. This section presents a divide and conquer approach which requires only $\log(d)$ executions of a F_i-minimum cut algorithm.

Lemma 1. Let F_i, F_j be distinct faces of μ, $i<j$. Let p be any F_j-minimum cut-cycle of $D(N)$ such that the closed region R bounded by p contains s. Then there exists an F_i-minimum cut-cycle q contained entirely in R.

Proof. Let q be any F_i-minimum cut-cycle. Let q' be the cut-cycle derived from q by repeatedly replacing subpaths connecting faces traversed by μ with the appropriate subpaths of μ (only apply replacements for which the resulting q' is cut-cycle).

Observe $c(q') \leq c(q)$ (else we can show μ is not a $\mu(s,t)$ path). Let R' be the closed region bounded by q. Suppose $R' \not\subseteq R$. Then there must be a subpath q_1 of q' connecting faces F^a, F^b of p such that q_1 only intersects R at F^a and F^b. Let p_1 be the subpath of p connecting F^a and F^b in R'. We claim $c(p_1) \leq c(q_1)$. Suppose $c(p_1) > c(q_1)$. By our construction of q', either q_1 avoids F_j, $F_j = F^a$ or $F_j = F^b$. In any case, we may derive a cut-cycle p' from p by substituting q_1 for p_1.

But this implies $c(p') < c(p)$, contradicting our assumption that p is a F_i-minimum cut-cycle.

Now substitute p_1 for q_1 in q'. The resulting cut-cycle is no more costly than q', since $c(p_1) \le c(q_1)$.

The lemma follows by repeated application of this process. □

The above lemma implies a method for dividing the planar standard network N, given an s-t cut X. Let N_X be the network derived from N by deleting all edges of X. N_X can be partitioned into two networks N_s, N_t, where no vertex of N_s has a path to t, and no vertex of N_t has a path to s. Also, each edge $c \in X$ must have connections to a vertex of N_s and a vertex of N_t.

Let N_s' be the planar network consisting of N_s, a new vertex t', and for each $e \in X$, add a new edge with cost $c(e)$ connecting t' to the vertex of e contained in N_s. Similarly, let N_t' be the planar network consisting of N_t, a new vertex s', and adding a new edge of cost $c(e)$ connecting s' to the vertex of e contained in N_t, for each $e \in X$. (Note that N_s' and N_t' are not necessarily standard since they may contain multiple edges connecting a given vertex to s or t.) Let DIVIDE(N,X,s) and DIVIDE(N,X,t) be the planar standard networks derived from N_s', N_t' respectively by merging multiple edges and setting the cost of each resulting edge to be the sum of the costs of the multiple edges from which it was derived.

Let E be the edges of network N.

Let Y be a set of edges of N_s (or N_t).

Let E(Y) be the set of edges of E derived from Y by substituting for any edge e connecting t' (or s') the corresponding edges of X from which e was derived.

The following theorem follows immediately from the above lemma and Proposition 3.

Theorem 4. Let X be an s-t cut of planar standard network N such that $D(X)$ is a F-minimum cut-cycle, for some face F in a $\mu(s,t)$ path of $D(N)$. Let X_s be a minimum s-t' cut of DIVIDE(N,X,s) and let X_t be a minimum s'-t cut of DIVIDE(N,X,s). Then $E(X_s)$ or $E(X_t)$ is a minimum s-t cut of N.

6. The Min s-t Cut Algorithm for Planar Networks

Theorem 4 of the previous Section 4 yields a very simple, but efficient, "divide and conquer" algorithm for computing minimum s-t cut of a planar standard network.

We assume the [Ford and Fulkerson, 1956] Algorithm (given in Section 3).

(i) (s,t)-PLANAR-MIN-CUT(N)

which computes a minimum s-t cut of (s,t)-planar standard network N in time $O(Q_L(n))$. We also assume algorithms (given in Section 4).

(ii) $\mu(s,t)$ PATH(D(N))

computes a $\mu(s,t)$ path of $D(N)$ in time $O(Q_L(n))$.

(iii) F-MIN-CUT-CYCLE(N,F_i,μ)

computes a F_i-minimum cycle of N (for F_i in $\mu(s,t)$ path μ), in time $O(Q_L(n))$.

Recursive Algorithm PLANAR-MIN-CUT(N,μ)

> input planar standard network $N = (G,c,s,t)$, where $G = (V,E)$, and
> $\mu(s,t)$ path μ.
> begin
>   Let $F_1,\ldots,F_d$ be the faces traversed by μ.
>   if $d = 1$ then return (s,t)-PLANAR-MIN-CUT(N);
>   else begin
> $X \leftarrow D^{-1}$(F-MIN-CUT-CYCLE($N,F_{\lfloor d/2 \rfloor},\mu$));
> $N_0 \leftarrow$ DIVIDE(N,X,s);
> $N_1 \leftarrow$ DIVIDE(N,X,t);
>   Let μ_0 (μ_1) be the subpath of μ contained in N_0
>   (respectively, N_1);

```
X_0 ← PLANAR-MIN-CUT(N_0,μ_0)
X_1 ← PLANAR-MIN-CUT(N_1,μ_1)
if c(E(X_0)) ≤ c(E(X_1))
     then return E(X_0)
     else return E(X_1);
end;
end;
```

For any $\omega \in \{0,1\}^r$, $r \geq 0$, inductively let $N_\omega = (G_\omega, c_\omega, s_\omega, t_\omega)$ be the planar standard network and let μ_ω be the $\mu(s_\omega,t_\omega)$-path in N_ω, defined by recursive calls to PLANAR-MIN-CUT. Suppose PLANAR-MIN-CUT(N_ω,μ_ω) is called. If μ_ω contains only one face, then let $N_{\omega 0}$ and $N_{\omega 1}$ be empty networks, and let $\mu_{\omega 0}$ and $\mu_{\omega 1}$ be empty paths. Else let X_ω be the s_ω-t_ω cut of N_ω computed by the call to D^{-1}(F-MIN-CUT-CYCLE(-1)) and let $N_{\omega 0}$, $N_{\omega 1}$ be the planar standard networks constructed by the calls to DIVIDE, and let $\mu_{\omega 0}$, $\mu_{\omega 1}$ be the subpaths of μ contained in $N_{\omega 0}$, $N_{\omega 1}$. Then it is easy to verify that $\mu_{\omega 0}$ is a $\mu(s_{\omega 0},t_{\omega 0})$-path in $N_{\omega 0}$ and $\mu_{\omega 1}$ is a $\mu(s_{\omega 1},t_{\omega 1})$-path in $N_{\omega 1}$. Furthermore, if d is the length of μ (the $\mu(s,t)$ path of N), there can be no more than $\log(d) = O(\log(n))$ recursive calls (where n is the number of vertices of N).

Let n_ω be the number of vertices of N_ω. Since N_ω is planar, the number of edges of N_ω is $6n_\omega - 12$ by Euler's Theorem.

Lemma 2. For any $r \geq 0$,

$$\sum_{\omega \in \{0,1\}^r} n_\omega = O(n) .$$

Proof. Suppose for some fixed $r_0 > 0$, this holds for all r, $0 \le r < r_0$. Consider some $\omega \in \{0,1\}^r$. Note that each edge of $N_{\omega 0}$ and $N_{\omega 1}$ constructed by DIVIDE corresponds to an edge of N_ω. Consider some fixed edge e of N_ω. Note that e appears only at most once in each of $N_{\omega 0}$ and $N_{\omega 1}$. If $e \notin X_\omega$ then e doesn't appear at all in one of $N_{\omega 0}$ or $N_{\omega 1}$. However if $e \in X_\omega$ then e may appear in both $N_{\omega 0}$ and $N_{\omega 1}$.

But (due to the merging of multiple edges in the definition of DIVIDE), for each $r_1 \ge r_0$, e appears in at most one $N_{\omega 0 \alpha}$ for any $\alpha \in \{0,1\}^{r_1}$ and not at all in $N_{\omega 0 \alpha'}$ for any $\alpha' \in \{0,1\}^{r_1} - \alpha$. Similarly, e appears in at most one $N_{\omega 1 \beta}$ for some $\beta \in \{0,1\}^{r_1}$. Thus by induction,

$$\sum_{\omega \in \{0,1\}^{r_0}} n_\omega = O(n) \quad . \qquad \square$$

We have shown:

Theorem 5. Given a planar standard network $N = (G,c,s,t)$ with $L = \mathrm{range}(c)$, and μ is a $\mu(s,t)$ path of N then PLANAR-MIN-CUT(N,μ) computes a minimum s-t cut of N in time $O(Q_L(n)\log(n))$.

By known upper bounds on the cost of maintaining queues (as discussed in the Introduction), we also have:

Corollary 5. A minimum s-t cut of N is computed in time $O(n \log^2(n))$ for general L (i.e., a set of positive reals), in time $O(n \log(n) \log\log(n))$ for the case L is a set of positive integers bounded by a polynomial in n, and in time $O(n \log(n))$ for the case $L = \{1\}$ (in this case N is a graph with identically weighted edges).

7. Conclusion

We have presented an algorithm for computing a minimum s-t cut of a planar undirected network. Our algorithm runs in an order of magnitude less time than previous algorithms for this problem. An additional attractive feature of this algorithm is its *simplicity*, as compared to certain other algorithms for computing minimum s-t cuts for sparse networks, [Galil, Naamad, 1979] and [Shiloach, 1978].

BIBLIOGRAPHY

[Aho, Hopcroft, and Ullman, 1974]

A. Aho, J. Hopcroft, J. Ullman, *The Design and Analysis of Computer Algorithms*, Addison Wesley, Reading, Mass (1974).

[Berge and Ghoula-Honri, 1965]

C. Berge and A. Ghouila-Honri, *Programming, Games, and Transportation Networks*, Methuen, Agincourt, Ontario, 1965.

[Boas, Kaas, and Zijlstra, 1977]

P. van Emde Boas, R. Kaas, E. Zijlstra, "Design and implementation of an efficient priority queue," *Mathematical Systems Theory*, 10, pp. 99-127 (1977).

[Cheston, Probert, Saxton, 1977]

G. Cheston, R. Probert, C. Saxton, "Fast Algorithms for Determination of Connectivity Sets for Planar Graphs," Univ. Saskatchewan, Dept. Comp. Science, Dec. 1977.

[Dijkstra, 1959]

E. Dijkstra, "A note on two problems in connections with graphs," *Numerische Mathematik*, 1, pp. 269-271 (1959).

[Even and Tarjan, 1975]

S. Even and R. Tarjan, "Network flow and testing graph connectivity," SIAM *J. Computing*, Vol. 4, No. 4, pp. 507-518 (Dec. 1975).

[Ford and Fulkerson, 1956]

C. Ford and D. Fulkerson, "Maximal flow through a network," *Canadian J. Math.*, 8, pp. 399-404 (1956).

[Ford and Fulkerson, 1962]

C. Ford and D. Fulkerson, *Flows in Networks*, Princeton University Press, Princeton, N. J., 1962

[Galil and Naamad, 1979]

Z. Galil and A. Naamad, "Network Flow and Generalized Path Compression," *Proceedings of Symposium of Theory of Computing*, Atlanta, Georgia, 1979.

[Gomory and Hu, 1961]

R. Gomory and T. Hu, "Multi-terminal network flows," SIAM *J. Appl. Math.*, pp. 551-570 (1961).

[Itai and Shiloach, 1979]

A. Itai and Y. Shiloach, "Maximum flow in planar networks," SIAM *J. of Computing*, Vol. 8, No. 2, pp. 135-150 (May 1979).

[Shiloach, 1978]

Y. Shiloach, "An $O(n\ I \cdot \log^2 I)$ Maximum-flow Algorithm," Comp. Science Dept., Stanford Univ., Stanford, Cal. (Dec. 1978).

[Shiloach, 1980]

Y. Shiloach, "A Multi-terminal minimum cut algorithm for planar graphs," SIAM *J. Computing*, Vol. 9, No. 2, pp. 214-219 (May 1980).

DATE

ILMED

8